UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/829,668	04/22/2004	Douglas C. Burger	119832-163869	6831

60172 7590 10/13/2009

SCHWABE, WILLIAMSON & WYATT, P.C.
1420 FIFTH, SUITE 3010
SEATTLE, WA 98101

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	**PAPER NUMBER**
2183	

MAIL DATE	DELIVERY MODE
10/13/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/829,668	BURGER ET AL.	
	Examiner	Art Unit	
	Robert Fennema	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 June 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *37-44,46-50 and 52-60* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *37-44,46-50 and 52-60* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *7/2/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 37-44, 46-50, and 52-60 are pending. Claims 37, 43-44, 46-50, 52-57, and 59-60 amended as per Applicant's request.

2. Examiner notes the entry of the following papers:

- Amendment filed 6/24/2009

- IDS filed 7/2/2009

Claim Objections

3. In Claim 57, Line 9, "subset" and "of" should be separated by a space.

4. In Claim 57, Line 11, "the subset of the second instruction" should read "the subset of the second instructions".

5. In Claim 60, Line 4, "wherein individual of the interconnected computing nodes" does not make sense. Examiner believes the claim is intended to read either "wherein each individual interconnected computing node" or "wherein individual interconnected computing nodes". Examiner is interpreting the claim as the former, but correction is required.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 59-60 are rejected under 35 U.S.C. 102(b) as being anticipated by Requa et al. ("The Piecewise Data Flow Architecture: Architectural Concepts", herein Requa).

8. As per Claim 59, Requa teaches: An apparatus, comprising:

computing resource including an execution unit configured to execute instructions (Page 427, Figure 1, the Scalar, Memory, or SIMD Processor); and

interconnect resource coupled to the computing resource to enable the apparatus to be a member of a group of interconnected computation nodes preselected to execute a group of instructions by successively executing subgroups of the group of instructions (Page 427, Second Column, Second paragraph, all processors/nodes are connected to each other through an interconnection network. Page 433 discloses successive execution of parts of a block);

wherein the interconnect resource includes:

at least one input port capable of being coupled to at least a first other preselected computation node included in the plurality of interconnected preselected computation nodes, the input port to receive input data (Page 427, Second Column,

Second paragraph, every node is connected to every other node, requiring an input port),

a first store coupled to the at least one input port to store the input data (Page 427, Second Column, Second paragraph, the FIFO queue),

a second store coupled to the execution unit, the second store to receive and store at least one instruction of a subgroup, the second store being a part of a frame of buffers spanning the plurality of interconnected preselected computation nodes to store a subgroup of instructions loaded into the frame of buffers prior to the subgroup of instructions having all necessary associated operations for execution (Page 426, first column, second paragraph, also see Page 433, “The PDF Block Processor”. Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor, and the instructions do not necessarily have their operands when first entered into the buffer),

an instruction wakeup unit to match the input data to the at least one stored instruction (PDF, paragraph 1, the operand source fields are modified as data comes in),

at least one output port coupled to the execution unit and capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Page 427, Second Column, Second paragraph, every node is connected to every other node, requiring an output port), and

a router to direct an output data of the execution unit from the at least one output port to the at least one preselected other computation node (Page 427, the FIFO queue also acts as a kind of router).

9. As per Claim 60, Requa teaches: A system, comprising:

a plurality of interconnected computing nodes configured to be pre-selectable to cooperatively execute a group of instructions (Page 427, Figure 1, the Scalar, Memory, or SIMD Processors);

wherein individual of the interconnected computing nodes includes:

computing resource including an execution unit configured to execute instructions (Page 427, Figure 1, the Scalar, Memory, or SIMD Processor); and

interconnect resource coupled to the computing resource to enable the computing node to cooperate with the other interconnected computation nodes to execute the group of instructions by successively executing subgroups of the group of instructions (Page 427, Second Column, Second paragraph, all processors/nodes are connected to each other through an interconnection network. Page 433 discloses successive execution of parts of a block);

wherein the interconnect resource includes:

at least one input port capable of being coupled to at least a first other preselected computation node including in the plurality of interconnected preselected computation nodes, the input port to receive input data (Page 427, Second Column,

Second paragraph, every node is connected to every other node, requiring an input port),

a first store coupled to the at least one input port to store the input data (Page 427, Second Column, Second paragraph, the FIFO queue),

a second store coupled to the execution unit, the second store to receive and store at least one instruction of a subgroup, the second store being a part of a frame of buffers spanning the plurality of interconnected preselected computation nodes to store a subgroup of instructions loaded into the frame of buffers prior to the subgroup of instructions having all necessary associated operands for execution (Page 426, first column, second paragraph, also see Page 433, “The PDF Block Processor”. Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor, and the instructions do not necessarily have their operands when first entered into the buffer),

an instruction wakeup unit to match the input data to the at least one stored instruction (PDF, paragraph 1, the operand source fields are modified as data comes in),

at least one output port coupled to the execution unit and capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Page 427, Second Column, Second paragraph, every node is connected to every other node, requiring an output port), and

a router to direct an output data of the execution unit from the at least one output port to the at least one preselected second other computation node (Page 427, the FIFO queue also acts as a kind of router).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 37-39, 43-44, 46-48, 50, and 52-58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa, in view of Patterson et al. ("Computer Architecture, A Quantitative Approach", herein Patterson).

12. As per Claim 37, Requa teaches: A method, comprising:

assigning a group of instructions selected from the plurality of groups of instructions partitioned from a program to a subset of interconnected computation nodes preselected from a plurality of interconnected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors); and

executing the subset of instructions as each one of the instructions in the subset of instructions loaded into the frame of buffers receives associated operands for execution (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed. Also see Page 435, Instruction Issue section. When the

instruction in the list has all required operands, it is sent to the instruction-issue section, and then a processor), but fails to teach:

loading a subset of instructions of the assigned group of instructions into a frame of buffers comprising stores disposed on the subset of interconnected preselected computation nodes having been assigned the group of instructions, prior to the subset of instructions having all necessary associated operands for execution.

Requa teaches loading instructions into a centralized instruction buffer, and only when all operands are available for execution, is it sent to the appropriate processor for execution, thus, Requa does not teach a buffer for each node, receiving instructions prior to the operands for said instructions being present. However, Patterson teaches of two methods to allow for dynamic scheduling, which has the advantage of removing multiple hazards: scoreboards and reservation stations. Requa's system is essentially a scoreboard, a centralized buffer that keeps track of when instruction operands are available in the register file, then issues when appropriate. However, Patterson also suggests a reservation station, which has additional advantages in that it removes additional WAW and WAR hazards (Page 252), by using register renaming, and furthermore, distributes the logic by having a reservation station in each functional unit, instead of in one centralized location (Page 252), and further does not require data to be written into the register file before it can be accessed by the instruction. The advantages of the distribution are explained on Page 257 of Patterson, where by having each functional unit (processor) having its own reservation station, when results are broadcast, multiple instructions can be issued at once, wherein a centralized issue

buffer cannot do so, in addition to the other advantages listed above. Therefore, given the aforementioned advantages, one of ordinary skill in the art would have been motivated to implement reservation station logic in Requa's architecture, which would involve both distribution of instruction buffers into each functional unit, and the issuing of instructions to those buffers prior to the operands being available.

13. As per Claim 38, Requa teaches: The method of claim 37, wherein at least one computation node included in the plurality of interconnected preselected computation nodes has at least one input port capable of being coupled to at least one preselected first other computation node included in the plurality of interconnected preselected computation nodes (Figure 1),

the input port to receive input data (Page 427, second column, Paragraph 2 (any consumer can receive any data), also see Page 433, The PDF Block Processor (herein referred to as PDF for the remainder of this claim, as this section will be referenced several times). Additionally, one would recognize a processor/execution unit inherently requires an input port to receive data),

a first store coupled to the at least one input port to store the input data (PDF, paragraph 1, input operands are stored in registers),

a second store coupled to an instruction sequencer, the second store to receive and store the at least one instruction (PDF, Paragraph 1, the instruction issue section holds instructions prior to execution),

an instruction wakeup unit to match the input data to the at least one instruction (PDF, paragraph 1, the operand source fields are modified as data comes in), at least one execution unit to execute the at least one instruction using the input data to produce output data (PDF, paragraph 1, the instructions are executed after receiving inputs),

at least one output port capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Figure 1 and Page 427, see below), and

a router to direct the output data from the at least one output port to the at least one preselected second other computation node (Page 427, second column, paragraph 2. Any consumer can receive any data from the interconnection network, thus all processors are capable of sending data to any other processor).

14. As per Claim 39, Requa teaches: The method of claim 37, wherein at least one of the plurality of groups of instructions is a basic block (Page 426, first column third paragraph).

15. As per Claim 43, Patterson teaches: The method of claim 37, wherein loading the group of instructions into a frame of buffers comprising stores disposed on the subset of interconnected preselected computation nodes includes:

sending at least two instructions selected from the group of instructions from an instruction sequencer to a selected computation node included in the subset of interconnected preselected computation nodes for storage in a store of the selected

computation node, prior to the at least two instructions having all necessary associated operands for execution (Page 252).

16. As per Claim 44, Requa teaches: The method of claim 37, wherein executing the subset of instructions loaded into the frame of buffers as each one of the instructions in the subset of instructions receives associated operands for execution includes:

matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node included in the subset of interconnected preselected computation nodes (Page 427, second column, second paragraph, where any consumer (processor) can receive any data, and instructions waiting to execute wait for results from the processor before executing).

17. As per Claim 46, Requa teaches: The method of claim 37, further comprising concurrently assigning another group of instructions selected from the plurality of groups of instructions to another or the same subset of interconnected preselected computation nodes for concurrent execution using one or more other frames of buffers comprising stores disposed on the another or same subset of interconnected preselected computation nodes (Page 436, Second Column, Second and Third paragraphs. Requa discusses that blocks can overlap each other as long as they do not need results from each other. Additionally, in the third paragraph, Requa discusses that the PDF architecture is capable of supporting multiple program executions simultaneously, which would also read on the limitation):

wherein the two groups of instructions are capable of concurrent execution (Page 436, Second Column, Second and Third paragraphs. Requa discusses that blocks can overlap each other as long as they do not need results from each other. Additionally, in the third paragraph, Requa discusses that the PDF architecture is capable of supporting multiple program executions simultaneously, which would also read on the limitation).

18. As per Claim 47, Requa teaches: An article comprising a machine-accessible medium having machine executable instructions stored therein, if executed, enable a machine to:

loading a subset of a group of instructions selected from a plurality of groups of instructions partitioned from a program to a frame of buffers (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor". Specifically, see Page 434, Figure 9. The Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor), and assigned to;

to execute the subset of instructions (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed), but fails to teach:

a frame of buffers comprising stores disposed on a subset of interconnected computation nodes preselected from a plurality of interconnected preselected computation nodes, and

wherein the subset of the group of instructions is loaded into the frame of buffers prior to having all necessary associated operands for execution.

Requa teaches loading instructions into a centralized instruction buffer, and only when all operands are available for execution, is it sent to the appropriate processor for execution, thus, Requa does not teach a buffer for each node, receiving instructions prior to the operands for said instructions being present. However, Patterson teaches of two methods to allow for dynamic scheduling, which has the advantage of removing multiple hazards: scoreboards and reservation stations. Requa's system is essentially a scoreboard, a centralized buffer that keeps track of when instruction operands are available in the register file, then issues when appropriate. However, Patterson also suggests a reservation station, which has additional advantages in that it removes additional WAW and WAR hazards (Page 252), by using register renaming, and furthermore, distributes the logic by having a reservation station in each functional unit, instead of in one centralized location (Page 252), and further does not require data to be written into the register file before it can be accessed by the instruction. The advantages of the distribution are explained on Page 257 of Patterson, where by having each functional unit (processor) having its own reservation station, when results are broadcast, multiple instructions can be issued at once, wherein a centralized issue buffer cannot do so, in addition to the other advantages listed above. Therefore, given the aforementioned advantages, one of ordinary skill in the art would have been motivated to implement reservation station logic in Requa's architecture, which would involve both distribution of instruction buffers into each functional unit, and the issuing of instructions to those buffers prior to the operands being available.

19. As per Claim 48, Requa teaches: The article of claim 47, wherein the machine executable instructions, if executed, further enable the machine to partition the program into the plurality of groups of instructions during compilation of the program (Page 429, first column, "PDF Architecture").

20. As per Claim 50, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions, if executed, enable the machine to:

statically assign each of the plurality of groups of instructions to a subset of interconnected preselected computation nodes preselected from a plurality of interconnected computation nodes for execution (Page 432, see Figure 8, and column 1, paragraph 2).

21. As per Claim 52, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions, if executed, enable the machine to:

generate a wakeup token to reserve an output data channel to connect selected computation nodes included in the subset of preselected interconnected preselected computation nodes (Page 427, section column, second paragraph).

22. As per Claim 53, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions, if executed, enable the machine:

to repeat said loading until the entire group of instructions are executed (Page 433, in order to execute a block, this clearly has to occur), and

to detect execution termination of the group of instructions including an output having architecturally visible data (Page 433, second column first paragraph, a flag is set when an execution is done, also see Page 430, second paragraph); and

committing the architecturally visible data to a register file (Page 430, second paragraph).

23. As per Claim 54, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions, if executed, enable the machine to repeat said loading until the entire group of instructions are executed (Page 433, in order to execute a block, this clearly has to occur), and to detect execution termination of the group of instructions including an output having architecturally visible data (Page 433, second column first paragraph, a flag is set when an execution is done, also see Page 430, second paragraph); and

committing the architecturally visible data to a memory (Page 430, second paragraph).

24. As per Claim 55, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions, in executed, enable the machine to route an output datum arising from executing one of the subset of instructions to a consumer node included in the plurality of interconnected preselected computation nodes, wherein the address of the consumer node is included in a token associated with

at least one instruction included in the subset of instructions (Page 429, second column, second paragraph).

25. As per Claim 56, Requa teaches: The method of claim 37 further comprising repeating said loading and executing until the entire group of instructions have been executed (Page 433, in order to execute a block, this clearly has to occur).

26. As per Claim 57, Requa teaches: An apparatus, comprising:

a plurality of interconnected computation nodes (Page 427, Figure 1, the Scalar, Memory, and SIMD Processors); and

storage medium coupled to the processor and having first instructions stored therein to be executed by the processor (Page 427, Main memory), wherein the first instructions are configured to

assign a group of second instructions selected from a plurality of groups of second instructions partitioned from a program to a preselected subset of the plurality of interconnected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors); and

wherein the subset of second instructions are executed as each one of the instructions in the subset of second instructions loaded into the frame of buffers receives associated operands for execution (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed. Also see Page 435, Instruction

Issue section. When the instruction in the list has all required operands, it is sent to the instruction-issue section, and then a processor), but fails to teach:

causing a subset of the second instructions of the assigned group of second instructions to be loaded into a frame of buffers comprising stores disposed on the subset of interconnected computation nodes having been assigned the group of second instructions, prior to the subset of the second instruction having all necessary associated operands for execution.

Requa teaches loading instructions into a centralized instruction buffer, and only when all operands are available for execution, is it sent to the appropriate processor for execution, thus, Requa does not teach a buffer for each node, receiving instructions prior to the operands for said instructions being present. However, Patterson teaches of two methods to allow for dynamic scheduling, which has the advantage of removing multiple hazards: scoreboards and reservation stations. Requa's system is essentially a scoreboard, a centralized buffer that keeps track of when instruction operands are available in the register file, then issues when appropriate. However, Patterson also suggests a reservation station, which has additional advantages in that it removes additional WAW and WAR hazards (Page 252), by using register renaming, and furthermore, distributes the logic by having a reservation station in each functional unit, instead of in one centralized location (Page 252), and further does not require data to be written into the register file before it can be accessed by the instruction. The advantages of the distribution are explained on Page 257 of Patterson, where by having each functional unit (processor) having its own reservation station, when results are

broadcast, multiple instructions can be issued at once, wherein a centralized issue buffer cannot do so, in addition to the other advantages listed above. Therefore, given the aforementioned advantages, one of ordinary skill in the art would have been motivated to implement reservation station logic in Requa's architecture, which would involve both distribution of instruction buffers into each functional unit, and the issuing of instructions to those buffers prior to the operands being available.

27. As per Claim 58, Requa teaches: The apparatus of claim 57, wherein at least one of the plurality of groups of second instructions is a selected of one a basic block, a hyperblock or a superblock (Page 426, first column third paragraph).

28. Claims 40-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa and Patterson, in view of Official Notice.

29. As per Claim 40, Requa teaches: The method of claim 37, but fails to teach: wherein at least one of the plurality of groups of instructions is a hyperblock.

Requa teaches of a system which uses basic blocks, but does not teach that the groups may be hyperblocks. However, Examiner is taking Official Notice that one of ordinary skill in the art would be capable and motivated to use hyperblocks in lieu of basic blocks, to take advantage of the ability to have multiple exits from a block (While Applicant has not provided a definition of a hyperblock, Examiner has found it to represent a block with one entrance and potentially (but not necessarily) more than one

exit).

30. As per Claim 41, Requa teaches: The method of claim 37, but fails to teach:

wherein at least one of the plurality of groups of instructions is a superblock.

Requa teaches of a system which uses basic blocks, but does not teach that the groups may be superblocks. However, Examiner is taking Official Notice that one of ordinary skill in the art would be capable and motivated to use superblocks in lieu of basic blocks, to take advantage of the ability to have multiple exits from a block (While Applicant has not provided a definition of a superblock, Examiner has found it to represent a block with one entrance and potentially (but not necessarily) more than one exit, however, Examiner is unclear how a superblock is different from a hyperblock).

31. Claims 42 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa and Patterson, in view of Fisher.

32. As per Claim 42, Requa teaches: The method of claim 37, but fails to teach:

wherein at least one of the plurality of groups of instructions is an instruction trace constructed by a hardware trace construction unit at run time.

While Requa teaches the method as disclosed in Claim 37, Requa does not teach about traces, or a trace construction unit to construct such a trace. However, Fisher teaches of Trace Scheduling, where the basic blocks used by Requa are compacted, and instead use traces (Page 462, Section D). The advantage to this

compaction method using traces allows for more efficient parallel code, done in a manner far more efficient than previous methods (Abstract). Given this advantage, one of ordinary skill in the art would have been motivated to use these traces in place of the basic blocks as taught by Requa to further increase the efficiency of the system.

33. As per Claim 49, Requa teaches: The article of claim 47, but fails to teach:

wherein the machine executable instructions, if executed, further enable the machine to partition the program into the plurality of groups of instructions is performed during run-time.

While Requa teaches the article as disclosed in Claim 47, Requa does not teach that the partitioning of the program is done by a trace mapper. However, Fisher teaches of Trace Scheduling, where the basic blocks used by Requa are compacted, and instead use traces (Page 462, Section D), created and optimized by a scheduler (Page 482, second column, second paragraph). The advantage to this compaction method using traces allows for more efficient parallel code, done in a manner far more efficient than previous methods (Abstract). Given this advantage, one of ordinary skill in the art would have been motivated to use these traces in place of the basic blocks as taught by Requa to further increase the efficiency of the system.

Response to Arguments

34. Applicant's arguments with respect to the rejection(s) of claim(s) 37-39, 43-44, 46-48, 50, and 52-58 under 35 U.S.C. 102(b) have been fully considered and are

persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Requa and Patterson. Examiner notes that Claims 59-60 do not require the instruction buffer to be present in the nodes, thus has not overcome the current rejection.

35. However, in regard to Applicant's argument that the processors of Requa are not interconnected to each other, Examiner disagrees, and asserts that they are interconnected to each other, as understood to those of ordinary skill in the art. If Applicant wishes to claim a particular layout of this interconnection, such as in Figure 2 of the instant application, that needs to be explicitly claimed, as the processors are all connected to each other, albeit indirectly, but Examiner does not believe the current claim language requires any narrower of a reading than that, if Applicant wishes for a narrower reading, then Examiner suggests an explicit recitation in the claims of the desired structure, for example, being arranged as a matrix, or with direct connections to each other. Additionally, as far as the Examiner can tell, the claims do not require that each processor/computation node be able to execute any instruction, simply that a subset of instructions are sent to a subset of nodes.

36. Applicant is welcome to contact the Examiner with any questions or concerns at the phone number listed below.

Conclusion

37. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert Fennema whose telephone number is (571)272-2748. The examiner can normally be reached on Monday-Thursday, 9:30-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert Fennema
Examiner
Art Unit 2183

RF

/David J. Huisman/
Primary Examiner, Art Unit 2183